BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

A LIMITED LIABILITY PARTNERSHIP
INCLUDING LAW CORPORATIONS

TELEPHONE (503) 439-8778

FACSIMILE (503) 439-6073

EMAIL: bstz_mail@bstz.com
www.bstz.com

INTELLECTUAL PROPERTY LAW

1925 NW AMBERGLEN PARKWAY,
SUITE 230
BEAVERTON, OR 97006

OTHER OFFICES

LOS ANGELES, CA
SILICON VALLEY / SUNNYVALE, CA
ORANGE COUNTY / COSTA MESA, CA
DENVER, CO
SEATTLE, WA

CLAIM AMENDMENTS

FOR ENABLING EXAMINER'S AMENDMENT

In response to the telephone conversation of August 27, 2009, please consider the following proposed amendments.

Amendments to the Claims are reflected in the listing of claims that begins on page 2 of this communication.

Remarks begin on page 15 of this communication.

Amendments to the Claims

This listing of amended claims is provided as a proposed amendment to the currently pending claims:

1. (Currently Amended) An integrated tracing and logging system employed within a network comprising:

a computer system having:

a processor coupled with a memory,

a tracing module associated with specified program code regions of an application, the tracing module to receive via an application programming interface (API) and process tracing method calls generated by the application when the specified program code regions are executed, a logging module separate from the

tracing module, the logging module associated with specified categories related to the network, the logging module to receive via the API and process logging method calls from network components associated with the categories, wherein the logging module and the tracing module are different respective subclasses of a controller class, wherein an initial configuring of the logging module and the tracing module includes both the logging module and the tracing module automatically inheriting from the controller class a first output destination to receive formatted messages, and

a formatter coupled to the tracing module and to the logging module, the formatter to receive tracing messages from the tracing module and logging messages from the logging module, the formatter including a configuration file storing a default format definition for the formatter, the formatter initially to format messages from the tracing module and the logging module according to the default format definition during a runtime of the formatter, wherein the formatter further to be

reconfigured during the runtime to format messages from the tracing module and the logging module according to a changed format definition, the reconfiguring the formatter including the configuration file receiving a change to the default format definition during the runtime, wherein the reconfiguring the formatter does not require recompiling of any source code of the integrated tracing and logging system, the formatter initially to automatically send formatted messages of the tracing module and logging module to the first output destination based on the automatic inheritance of the first output destination by the tracing module and logging module; and [[an]]

a second output destination ~~to receive the formatted output of the tracing module and the logging module~~ different from the first output destination, wherein a method call of one of the logging module and the tracing module is performed during the runtime to store in the configuration file data assigning the second output

destination to the one of the logging module and the tracing module, the formatter further to automatically send formatted output of the tracing module and the logging module to the second output destination based on the data assigning the second output destination to the one of the logging module and the tracing module.

2. (Original) The system of claim 1, wherein the formatter is one of a list formatter, a human-readable formatter, and a markup language formatter.

3. (Previously Presented) The system of claim 1, wherein the configuration file further defines one or more properties of the formatter.

4. (Original) The system of claim 3, wherein the configuration file includes an identifier to identify the formatter.

5. (Original) The system of claim 3, wherein the one or more properties are formatted as key-value-pair properties, each key-value pair having a key to specify an attribute and a value to provide a definition for the specified attribute.

6. (Previously Presented) The system of claim 3, wherein a format definition of the configuration file includes one or more fields.

7. (Currently Amended) The system of claim 6, wherein the one or more fields includes at least one of

a timestamp field to indicate a time for one of the received ~~message~~ tracing messages and logging messages;

a location of origin field to indicate a source of one of the received ~~message~~ tracing messages and logging messages;

a thread identifier field to indicate a thread associated with one of the received ~~message~~ tracing messages and logging messages;

a message severity indicator field to indicate a severity level of one of the received ~~message~~ tracing messages and logging messages; and

a message identifier field to identify one of the received ~~message~~ tracing messages and logging messages.

8. (Previously Presented) The method of claim 1, wherein the output destination is at least one of

a console;

a trace file; and

a log file.

9. (Canceled).

10. (Currently Amended) A computer-implemented method employed within a network comprising:

creating an instance of a tracing controller associated with specified program code regions of an application, the tracing controller instance to receive and process tracing method calls generated by the application when the specified program code regions are executed;

creating an instance of a logging controller associated with specified categories related to the network, the logging controller to receive and process logging method calls from network components associated with the categories, wherein the logging controller instance and the tracing controller instance are different respective subclasses of a controller class, wherein a default output

destination for formatted messages is automatically inherited from the controller class both by the logging controller instance and by the tracing controller instance;

providing a common application programming interface of the tracing controller instance and the logging controller instance, whereby the tracing controller instance and the logging controller instance are accessed;

creating an instance of a formatter coupled to the tracing controller instance and the logging controller instance, the formatter including a configuration file;

receiving at the formatter instance tracing messages from the tracing controller instance and logging messages from the logging controller instance, wherein the formatter instance is initially to automatically send formatted messages of the tracing controller instance and logging controller instance to the default output destination

based on the inheritance of the default output destination by the tracing controller instance and logging controller instance;

the formatter instance performing a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a default format definition stored in the configuration file;

after the first formatting, reconfiguring the formatter instance to format messages from the tracing controller instance and the logging controller instance according to a changed format definition, the reconfiguring during the runtime, the reconfiguring including:

changing the default format definition stored in the configuration file, and

performing a method call of one of the logging controller instance and the tracing controller instance, the method call to store in the configuration file data assigning a first output destination different from the default output destination to the one of the logging controller instance and the tracing controller instance,

wherein reconfiguring the formatter instance does not require a recompiling of any source code; [[and]]

after the reconfiguring, the formatter instance performing a second formatting of received messages from the one of the tracing module and the logging module, the second formatting according to the changed format definition; and

the formatter instance automatically sending an output of the second formatting to the first output destination, the sending based on the data assigning the

first output destination to the one of the logging controller instance and the tracing controller instance.

11. (Canceled).

12. (Currently Amended) The method of claim [[11]] 10, wherein ~~configuring the message format~~ changing the default format definition comprises providing an identifier to the configuration file to identify the selected formatter.

13. (Currently Amended) The method of claim 12, wherein ~~configuring the message format~~ changing the default format definition further comprises specifying one or more fields for the message format.

14. (Currently Amended) The method of claim 13, wherein specifying one or more fields comprises specifying at least one of

a timestamp field to indicate a time for one of the received ~~message~~ tracing messages and logging messages;

a location of origin field to indicate a source of one of the received ~~message~~ tracing messages and logging messages;

a thread identifier field to indicate a thread associated with one of the received ~~message~~ tracing messages and logging messages;

a message severity indicator field to indicate a severity level of one of the received ~~message~~ tracing messages and logging messages; and

a message identifier field to identify one of the received ~~message~~ tracing messages and logging messages.

15. (Currently Amended) The method of claim 10, further comprising:

providing a filter to the specified output destination to selectively filter one of the received ~~message~~ tracing messages and logging messages.

16. (Previously Presented) A system comprising:

a computer system having a processor coupled with a memory, the computer system further including,

a means for creating an instance of a tracing controller associated with specified program code regions of an application, the tracing controller instance to receive via an application programming interface (API) and process tracing method calls generated by the application when the specified program code regions are executed,

a means for creating an instance of a logging controller associated with specified categories related to the network, the logging controller to receive via the

API and process logging method calls from network components associated with the categories, wherein the logging controller instance and the tracing controller instance are different respective subclasses of a controller class, wherein an initial configuring of the logging controller instance and the tracing controller instance includes both the logging controller instance and the tracing controller instance automatically inheriting from the controller class a first output destination to receive formatted messages,

means for creating an instance of a formatter to receive tracing messages from the tracing controller instance and logging messages from the logging controller instance, the formatter instance including a configuration file, the formatter instance further to perform a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a format definition stored in

the configuration file, the formatter instance initially to automatically send formatted messages of the tracing controller instance and logging controller instance to the first output destination based on the automatic inheritance of the first output destination by the tracing controller instance and logging controller instance, and

means for reconfiguring the formatter instance after the first formatting and during the runtime of the formatter instance, the reconfiguring including changing the format definition stored in the configuration file, wherein the reconfiguring the formatter instance does not require a recompiling of any source code, wherein the reconfigured formatter instance to perform a second formatting of received messages from the tracing controller instance and the logging controller instance, the second formatting according to the changed format definition stored in the configuration file; and [[an]]

a second output destination ~~to receive the formatted output of the one of the tracing controller instance and the logging controller instance~~ different from the first output destination, wherein a method call of one of the logging controller instance and the tracing controller instance is performed during the runtime to store in the configuration file data assigning the second output destination to the one of the logging controller instance and the tracing controller instance, the formatter instance further to automatically send formatted output of the tracing controller instance and the logging controller instance to the second output destination based on the data assigning the second output destination to the one of the logging controller instance and the tracing controller instance.

Claims 17. (Canceled).

18. (Previously Presented) The system of claim 16, wherein the means for changing the format definition comprises:

a means for specifying one or more fields for a defined message format.

19. (Currently Amended) The system of claim 18, wherein the means for specifying one or more fields comprises a means for specifying at least one of

a timestamp field to indicate a time for one of the received ~~output~~ tracing messages and logging messages;

a location of origin field to indicate a source of one of the received ~~output~~ tracing messages and logging messages;

a thread identifier field to indicate a thread associated with one of the received ~~output~~ tracing messages and logging messages;

a message severity indicator field to indicate a severity level of one of the received ~~output~~ tracing messages and logging messages; and

a message identifier field to identify one of the received ~~output~~ tracing messages and logging messages.

20. (Previously Presented) An article of manufacture comprising:

an electronically accessible medium providing instructions that, when executed by an apparatus, cause the apparatus to

create an instance of a tracing controller associated with specified program code regions of an application, the tracing controller instance to receive and process tracing method calls generated by the application when the specified program code regions are executed;

create an instance of a logging controller associated with specified categories related to the network, the logging controller to receive and process logging method calls from network components associated with the categories, wherein the logging controller instance and the tracing controller instance are different respective subclasses of a controller class, wherein a default output destination for formatted messages is automatically inherited from the controller class both by the logging controller instance and by the tracing controller instance;

provide a common application programming interface of the tracing controller instance and the logging controller instance, whereby the tracing controller instance and the logging controller instance are accessed;

create an instance of a formatter coupled to the tracing controller instance and the logging controller instance, the formatter including a configuration file;

receive at the formatter instance tracing messages from the tracing controller instance and logging messages from the logging controller instance, wherein the formatter instance is initially to automatically send formatted messages of the tracing controller instance and logging controller instance to the default output destination based on the inheritance of the default output destination by the tracing controller instance and logging controller instance;

perform at the formatter instance a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a default format definition stored in the configuration file;

after the first formatting, reconfigure the formatter instance to format messages from the tracing controller instance and the logging controller instance

according to a changed format definition, the reconfiguring during the runtime, the reconfiguring including:

changing the default format definition stored in the configuration file, and

performing a method call of one of the logging controller instance and the tracing controller instance, the method call to store in the configuration file data assigning a first output destination different from the default output destination to the one of the logging controller instance and the tracing controller instance,

wherein reconfiguring the formatter instance does not require a recompiling of any source code; [[and]]

after the reconfiguring, the formatter instance performing a second formatting of received messages from the one of the tracing module and the logging module, the second formatting according to the changed format definition; and

automatically send an output of the second formatting from the formatter instance to the first output destination, the sending based on the data assigning the first output destination to the one of the logging controller instance and the tracing controller instance.

Claim 21. (Canceled).

22. (Previously Presented) The article of manufacture of claim 20, wherein the instructions that, when executed by the apparatus, cause the apparatus to change the format definition for the formatter, cause the apparatus to provide one or more fields for a defined message format.

23. (Previously Presented) An apparatus comprising:

an application; and

a processor and logic executable thereon to

create an instance of a tracing controller associated with specified program code regions of the application, the tracing controller instance to receive and process tracing method calls generated by the application when the specified program code regions are executed;

create an instance of a logging controller associated with specified categories related to a network, the logging controller instance to receive and process logging method calls from network components associated with the categories, wherein the logging controller instance and the tracing controller instance are different respective subclasses of a controller class, wherein an output destination is inherited from the

controller class both by the logging controller instance and by the tracing controller instance;

provide a common application programming interface of the tracing controller instance and the logging controller instance, whereby the tracing controller instance and the logging controller instance are accessed;

create an instance of a formatter coupled to the tracing controller instance and the logging controller instance, the formatter including a configuration file, wherein the formatter is one of a list formatter, a human-readable formatter, and a markup language formatter, wherein the formatter instance is initially to automatically send formatted messages of the tracing controller instance and logging controller instance to the default output destination based on the inheritance of the

default output destination by the tracing controller instance and logging controller instance;

receive at the formatter instance tracing messages from the tracing controller instance and logging messages from the logging controller instance~~, the receiving based on an assignment of the formatter instance to the output destination~~;

perform at the formatter instance a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a default format definition stored in the configuration file;

after the first formatting, reconfigure the formatter instance to format messages from the tracing controller instance and the logging controller instance

according to a changed format definition, the reconfiguring during the runtime, the reconfiguring including:

changing the default format definition stored in the configuration file, the changing the default format definition including specifying one or more fields for a defined message format, the one or more fields including at least one of

a timestamp field to indicate a time for the received message,

a location of origin field to indicate a source of the received message,

a thread identifier field to indicate a thread associated with the received message,

a message severity indicator field to indicate a severity level of the received message, and

a message identifier field to identify the received message, and performing a method call of one of the logging controller instance and the tracing controller instance, the method call to store in the configuration file data assigning a first output destination different from the default output destination to the one of the logging controller instance and the tracing controller instance,

wherein reconfiguring the formatter instance does not require a recompiling of any source code; [[and]]

after the reconfiguring, the formatter instance performing a second formatting of received messages from the one of the tracing module and the logging module, the second formatting according to the changed format definition; and

automatically send an output of the second formatting from the formatter instance to the first output destination, the sending based on the data assigning the

first output destination to the one of the logging controller instance and the tracing controller instance.

24. (Canceled).

25. (Original) The apparatus of claim 23, wherein the configuration file includes an identifier to identify the formatter.

Claims 26 through 28. (Canceled).

Remarks

This communication, and the listing of amended claims included herein, is in response to a request from Examiner Tuan A. Vu that Applicants provide a version of proposed claim amendments which are the result of discussions between Examiner Tuan A. Vu and Applicants' representative Dermot G. Miller (Reg. No. 58,309). These discussions – initiated by Examiner Tuan A. Vu on

August 10, 2009 – resulted in an August 27, 2009 agreement that US Patent Application Number 10/815,018 would be put in condition for allowance by entering as Examiner's Amendments certain claim amendments listed herein. The listing of amended claims, including the status identifiers of said claims, is presented herein solely to aid preparation of such Examiner's Amendments.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Date: August 27, 2009

Dermot G. Miller

Attorney for Applicants
Reg. No. 58,309
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
(503) 439-8778